cifbuild

February 23, 2011

Abstract

Construct CCF Calibration or Master Index Files.

1 Instruments/Modes

Instrument	Mode

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

The XMM-Newton calibration data base is a collection of algorithms (described in the Calibration Handbook [1]), and Current Calibration File (CCF) constituents [2]. The latter contain the numerical values driving the calibration algorithms.

The SAS tasks access the calibration data base through the Calibration Access Layer (CAL, **cal**). The CAL is pointed to the correct set of CCF constituents through a CCF Index File (CIF).

cifbuild is used to generate the CIF appropriate to a given observation and analysis date.

Details on how to use a CIF in conjunction with the SAS, and what environment variables must be used, can be found in the CAL documentation (see documentation of cal).

3.1 Where to find the CCF constituents

Information about the CCF constituents and where to download them can be found on the CCF web pages.

3.2 Generation of the Calibration Index File

As described in detail below 3.4 (see also [2]) the list of calibration datasets comprising a CIF is ruled by two dates: the observation date and the analysis date.

By default **cifbuild** reads the observation date from the Observation Data File (ODF) pointed to via the `SAS_ODF` environment variable.

If an ODF is not available the observation date can be specified with the parameters `withobservationdate` and `observationdate`.

The analysis date is given with the parameter `analysisdate`, and it defaults to `now`. Note that the analysis date can be any date (in the past or in the future): it is used to retrieve the CCF constituents applicable at the specified point in time.

For instance, to generate the CIF applicable on 2010-01-01 say `analysisdate=2010-01-01`. What CCF constituents would one have used on 2005-01-01? Use `analysisdate=2005-01-01`.

The analysis and observation dates used in the creation of the CIF are recorded in the two block attributes `OBSVDATE` and `ANALDATE` respectively. (See 7.)

Note that it is possible to use an analysis date earlier than the observation date. This is flagged with a warning, and may lead to an inconsistent calibration.

3.2.1 Using the CCF constituents

cifbuild can build a CIF by examining the CCF constituents found in a number of directories specified by the user. The list of directories (use the character `:` as a separator) should be specified through the environment variable `SAS_CCFPATH`. Alternatively, the same information can be specified on the command line (see the parameters `withccfpath` and `ccfpath`).

3.2.1.1 Advanced usage Normally a valid CCF constituent will have the `CATEGORY` attribute set to `XMMCCF`. However, it is possible to instruct **cifbuild** to accept constituents with a different category via the parameter `category`. For instance, the SOC distributes CCF constituents tuned to the output of the science simulator. These datasets have the `CATEGORY` attribute set to `SCISIMCCF`.

The category can also be ignored altogether with the parameter `ignorecategory`.

If the parameter `append` is true, then **cifbuild** appends to the CIF set pointed to by `calindexset` any other CCF constituent that is not already in that CIF. With this mechanism it is possible to build a CIF incrementally, giving priority to a certain category of constituents (see example 8).

3.2.2 Using a Master Index File

cifbuild can also be used to generate a CIF based on the contents of the Master Index File (MIF). The MIF contains the list of all CCF constituents released from the start of the mission until the date specified in the `SUBDATE` attribute. The generation of a CIF in this case requires that the user specify the MIF to be used with the parameters `withmasterindexset` and `masterindexset`.

The latest version of the MIF is available from the CCF area at the XMM-Newton Science Operation

Centre.

3.2.3 Examples

In reference to the examples below, see also the CAL documentation (see documentation of cal) on how to specify a CIF.

Note: in the following examples we use the Bourne shell syntax to specify environment variables. C shell users should convert that to `setenv VARIABLE value`.

1. This are the recommended steps for people who have access to a local collection of CCF constituents:

   (a) `export SAS_ODF=/path/to/odf_dir`[1]

   (b) `export SAS_CCFPATH=/path/to/ccfstore1:/path/to/ccfstore2`

   (c) `cifbuild`

   This creates the file `ccf.cif` (the default value of `calindexset`) in the current directory. The CCF constituents are searched among the files in the two directories specified in the environment variable `SAS_CCFPATH`. The observation date is taken from the active ODF as indicated by the variable `SAS_ODF`. The analysis date defaults to `now` (the default value of `analysisdate`).

   If the `usecanonicalname` had been set to `true`, the name of the CIF would have been of the form `rrrr_pppppppooll.cif`, where `rrrr` is the revolution number, and `pppppppooll` is the observation identifier.

2. This is the recommended mode of operation for those who want to generate a CIF from a MIF. It is not necessary to have access to any of the CCF constituents.

   ```
   cifbuild --withmasterindexset=yes --masterindexset=XMM_CALINDEX_0123.CCF
   ```

   The CIF is built from the list of CCF constituents specified in `XMM_CALINDEX_0123.CCF`.

3.
   ```
   cifbuild --withccfpath=yes
   --ccfpath="/path/to/ccfstore1/:/path/to/ccfstore2"
   --fullpath=yes
   ```

   This creates the file `ccf.cif` (the default value of `calindexset`) in the current directory. The CCF constituents are searched among the files in the two directories specified with `ccfpath`. The observation date is taken from the active ODF as indicated by the variable `SAS_ODF`. The analysis date defaults to `now` (the default value of `analysisdate`). The CIF contains the full path to the CCF constituents.

   To use the CIF set the environemnt variable `SAS_CCF` to be `ccf.cif`. If you set `SAS_CCF` to be the absolute path to the CIF, then the latter can be used from any other directory.

4.
   ```
   cifbuild --withccfpath=yes --ccfpath=/path/to/ccfstore
   ```

   The CIF `ccf.cif` contains the base name of the CCF constituents found in `/path/to/ccfstore`.

   To use the CIF place it in `/path/to/ccfstore` and set the environemnt variable `SAS_CCF` to be `/path/to/ccfstore`.

5.
   ```
   cifbuild --withccfpath=yes --ccfpath=/path/to/ccfstore
   --calindexset=/path/to/ccfstore/myobservation.cif
   ```

   As in 4, but now the CIF is called `myobservation.cif` and is written directly to the directory `/path/to/ccfstore`.

[1] If you generated a summary file with the task **odfingest**, then you can also say `export SAS_ODF=/path/to/summary_file`.

To use the CIF set the environment variable `SAS_CCF` to be `/path/to/ccfstore/myobservation.cif`.

6.
   ```
   cifbuild --withccfpath=yes
   --ccfpath="/path/to/ccfstore1/:/path/to/ccfstore2"
   --withobservationdate=yes --observationdate="2000-01-13T00:00:00"
   --fullpath=yes
   ```

   As in 3, but now the observation date is specified to be Midnight January 13, 2000.

7.
   ```
   cifbuild --withccfpath=yes --ccfpath="/path/to/ccfstore1/:/path/to/ccfstore2"
   --analysisdate="2000-03-19T00:00:00" --fullpath=yes
   ```

   As in 3, but the observation date is taken from the active ODF, but the CIF is built as if the analysis was being performed on March 19, 2000.

8.
   ```
   cifbuild --calindexset=ccf.cif --category=SCISIMCCF --ccfpath=/path/to/ccfstore/
   --withccfpath=yes
   ```

   ```
   cifbuild --calindexset=ccf.cif --withccfpath=yes --ccfpath=/path/to/ccfstore/
   --append=yes
   ```

   This builds a CIF that contains constituents of category SCISIMCCF if they exist, otherwise constituents of category `XMMCCF` are used.

3.3 Generation of a Master Index File

This section is not of interest for the general user. There is no reason to do this outside the XMM-Newton Science Operations Centre.

By setting the parameter `masterindex` to `yes` **cifbuild** can be instructed to generate a MIF.

In this mode of operations the command line value of `fullpath` is ignored.

All CCF constituents found in the directories specified via `ccfpath` (or `SAS_CCFPATH`) are listed in the output calibration index set (`calindexset`).

3.4 Excerpts from the Current calibration File Interface Control Document

This section is taken verbatim from [2].

3.5 File Naming Convention

The constituents of the current calibration file are named as follows:

scope_calname_issue.ccf

where:

scope indicates the scope of the calibration file (at most 5 characters long). The list of scope identifiers is given in table 1.

calname is the calibration data type identifier (at most 17 characters long).

issue is the file issue number (integer ≥ 0 and ≤ 9999, zero padded).

File names are not case sensitive. In this document they are type set with uppper and lower case letters for readability.

xmm	general files or files shared by more than one instrument
om	Optical Monitor files
xrt1, xrt2, xrt3	X-ray Telescope files
emos1, emos2	European Photon Imaging Camera MOS files
epn	European Photon Imaging Camera PN files
rgs1, rgs2	Reflection Grating Spectrometer files
erm	EPIC Radiation Monitor files

Table 1: Allowed scope identifiers

3.6 Membership of the Current Calibration File

Here we address the issue of how to determine what calibration files make up the current calibration file for a given observation time.

Each calibration file has a validity start date and a validity end date. These are indicated by the FITS keywords `VALDATE` and `EVALDATE`. A missing `EVALDATE` is taken to mean *no end of validity.*

Given the naming convention above, a calibration file can be labeled with a quadruplet of the form $\{Fname, Issue, T_{start}, T_{end}\}$, where $Fname \equiv$ **scope_calname**, and T_{end} is optional.

New issues of a calibration file can enter the archive with any T_{start}, any T_{end}, and increasing issue numbers.

In the simplest case (no end of validity date), a new version of the calibration file — call it $\{Fname, Issue', T'_{start}\}$, will have $T_{start} < T'_{start}$ and $Issue < Issue'$ (figure 1).

A more complex situation is illustrated in figure 2. Here new versions are introduced to improve on past calibrations.

In figure 3 we show an example where a CCF constituent has an end of validity date. In this case at the time T the situation is the same as shown in figure 2, whereas at the time $T1$ the constituent *xmm_ExampleTwo_005.ccf* is no longer applicable.

Within this scheme, at any time T the applicable issue of the calibration file $Fname$ is obtained by selecting among all files $\{Fname, Issue, T_{start} \leq T, T_{end} \geq T\}$ the one with the higher issue number.

Should it become necessary, a new $Fname$ can be introduced in the archive, as illustrated in figure 4.

With this scheme the number of constituents in the current calibration file is not fixed, and can be determined only when the time T is specified.

3.7 See also

Other SAS tasks to manipulate calibration index files are:

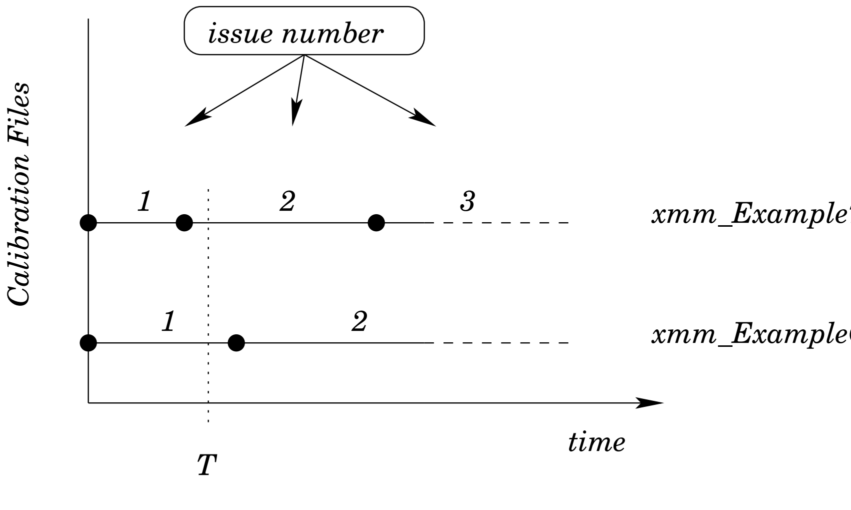

Figure 1: Current calibration file with two files. At the time T the current calibration file consists of *xmm_ExampleOne_0001.ccf* and *xmm_ExampleTwo_0002.ccf*

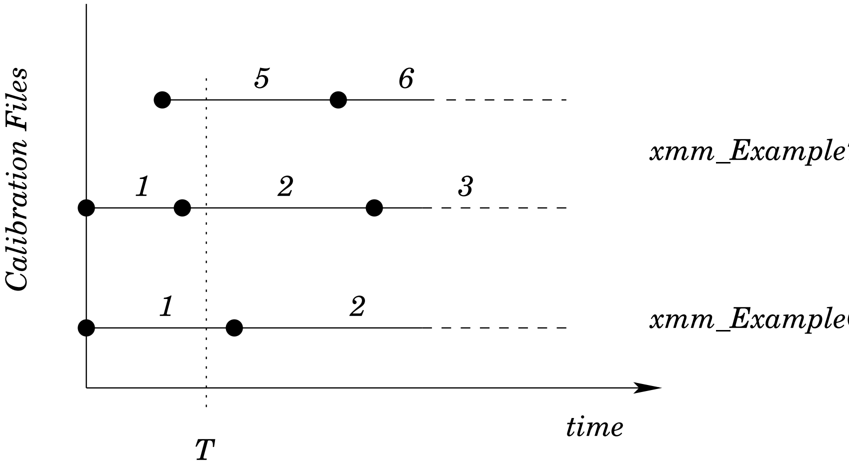

Figure 2: Current calibration file with two files: update. At the time T the current calibration file consists of *xmm_ExampleOne_0001.ccf* and *xmm_ExampleTwo_0005.ccf*

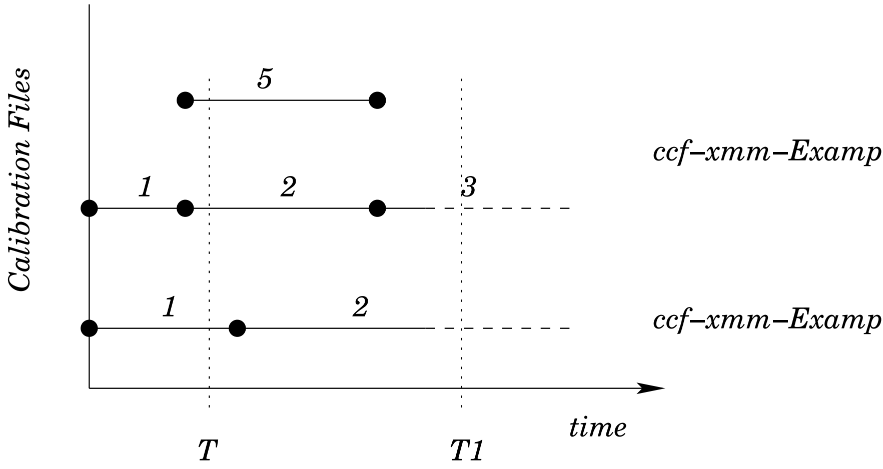

Figure 3: Current calibration file with two files: update. At the time T the current calibration file consists of *xmm_ExampleOne_0001.ccf* and *xmm_ExampleTwo_0005.ccf*. At the time $T1$ the current calibration file consists of *xmm_ExampleOne_0001.ccf* and *xmm_ExampleTwo_0003.ccf*.

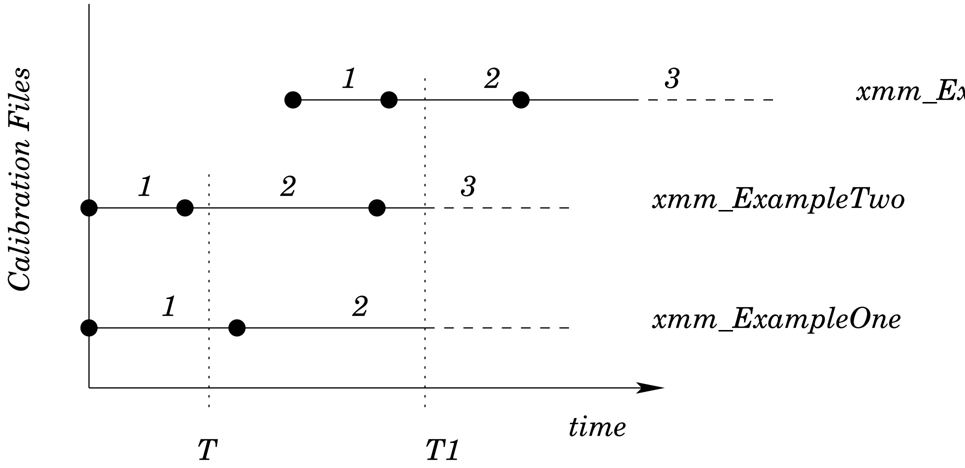

Figure 4: Current calibration file with three files. At the time $T1$ the current calibration file consists of *xmm_ExampleOne_0002.ccf* and *xmm_ExampleTwo_0003.ccf* and *xmm_ExampleThree_0002.ccf*

- **cifdiff**: list the difference between two CIFs.
- **cifremove**: remove CCF constituents from a CIF.
- **cifinsert**: insert CCf constituents into a CIF.

4 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints

calindexset	no	e	ccf.cif	

Name of the output calibration index set.

withccfpath	no	b	no	

Use the directories listed in ccfpath? Set to no means use `SAS_CCFPATH`.

usecanonicalname	no	b	no	yes | no

Autogenerate the CIF name based on the ODF identifier? If set to `yes`, if the observation date is read from an ODF, and if `withmasterindex` is set to `no`, the name of the CIF will be of the form `rrrr_pppppppooll.cif`, where `rrrr` is the revolution number, and `pppppppooll` is the observation identifier.

ccfpath	no	s	.	

Colon-separated list of directories where CCF constituents are to be looked for.

recurse	no	b	no	

Recursively look for CCF constituents in any subdirectory found in ccfpath or in `SAS_CCFPATH`?

fileglob	no	s	*.ccf|*.CCF	

File name glob patterns to match CCF constituents. The delimiter is |.

fullpath	no	b	no	

Write in the calibration index set the full path names of the CCF constituents? The valu of this parameter is ignored if `withmasterindexset` is set to `true`.

withobservationdate	no	b	no	

Use the user-specified observation date? Otherwise will read the data from the active ODF (environment variable `SAS_ODF`).

observationdate	no	w	now	

Date when the observation was made.

analysisdate	no	w	now	

Date when the analysis is performed.

category	no	s	XMMCCF|SCISIMCCF	

Category of constituents to look for. Use * for any category.

ignorecategory	no	b	no	

Pay no attention to the CATEGORY attribute?

masterindex	no	b	no	

Create a master calibration index set (aka MIF)?

withmasterindexset	no	no	b	

Make use of the master index set (aka MIF) instead of looking directly at the CCF constituents?

masterindexset	no	ccf.mif	e	

Name of the MIF to be used to evaluate the calibration index set (aka CIF).

append	no	no	b	

Append CCF constituents to an existing calindexset?

5 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

NoMatchingCcfConstituents *(warning)*

cifbuild did not find any CCF constituent matching the specified selection criteria.

corrective action: None.

PossibleInconsistentCalibration *(warning)*
The user specified `analysisdate` < `observationdate`. The resulting Calibration Index File may contain a set of CCF constituents that are not consistent and lead to the wrong calibration being applied.
corrective action: None.

MissingAttribute *(warning)*
cifbuild expects to find a number of attributes in each CCF constituent. One or more of these attributes were not found.
corrective action: If a critical attribute is missing the constituent may be discarded at a later stage.

InvalidCcfConstituent *(warning)*
cifbuild could not recognize a dataset as a valid CCF constituent.
corrective action: The dataset is ignroed.

TableNotFound *(error)*
cifbuild was asked to open a calibration index file, but the required table was not found.

NoCcfPath *(error)*
The environment variable `SAS_CCFPATH` is not set, and no CCF search path was given on the command line. See `withccfpath`.

6 Input Files

1. Calibration Master Index File: this is available in the CCF. Its format is the same as the CIF (see below).
2. CCF constituents [2].

7 Output Files

1. Calibration index file (either a CIF or a MIF)

   A binary table `CALINDEX` with the following columns:

   - `TELESCOP`, character(4): copied from the CCF constituent
   - `SCOPE`, character(6): calibration scope
   - `TYPEID`, character(32): calibration type
   - `ISSUE`, integer: issue number
   - `VALDATE`, character(19): start of validity date (yyyy:dd:mmZhh:mm:ss)
   - `VALDATE-END`, character(19): end of validity date (yyyy:dd:mmZhh:mm:ss). An empty value means that the constituent has no end of validity date.
   - `FNAME`, character(256): name of the constituent
   - `DATE`, character(19): creation date (yyyy:dd:mmZhh:mm:ss)
   - `FSIZE`, integer: constituent size in bytes
   - `SUBDATE`, character(19): submission date (yyyy:dd:mmZhh:mm:ss)
   - `EXTSEQU`, character(32): extension sequence

- EXTSEQID, character(256): extension sequence identifiers
- MD5, character(32): MD5 signature of the constituent
- CREATOR, character(64): copied from the CCF constituent

Two attributes are used to record the dates used in the creation of the CIF:

- OBSVDATE: the observation date.
- ANALDATE: the analysis date.

8 Algorithm

```
if(not withmasterindexfile) {

  foreach(directory is SAS_CCFPATH or in ccfpath) {
    foreach(file matching glob pattern) {
      if(is_a_ccf_file) {
         extract CCF attributes
         add to list organized by scope-type
      }
    }
    foreach(scope-type) {
      select constituent with:
          VALDATE <= observationdate
          SUBDATE <= analysisdate
          EVALDATE >= observationdate
          highest ISSUE
      write selected constituet to CIF
  }
} else { // use masterindexfile
  foreach(scope-type in MIF) {
    select constituent with:
        VALDATE <= observationdate
        SUBDATE <= analysisdate
        highest ISSUE
   }
}
```

9 Comments

References

[1] Christian Erd, Phillipe Gondoin, David Lumb, Rudi Much, Uwe Lammers, and Giuseppe Vacanti. Calibration Access and Data Handbook. Technical Report XMM-PS-GM-20, ESA/SSD, Jan 14 2000. Found at the URL: `http://xmm.vilspa.esa.es/docs/documents/CAL-MAN-0001-2-1.ps.gz`.

[2] ESA. Interface control document for the XMM current calibration file. Technical Report XMM-GEN-ICD-0005, ESA/SSD, Dec 2001. Issue 4.0.